Quebra de autenticação e Gerenciamento de Sessão.

Marco Antônio Diniz dos Santos Reis

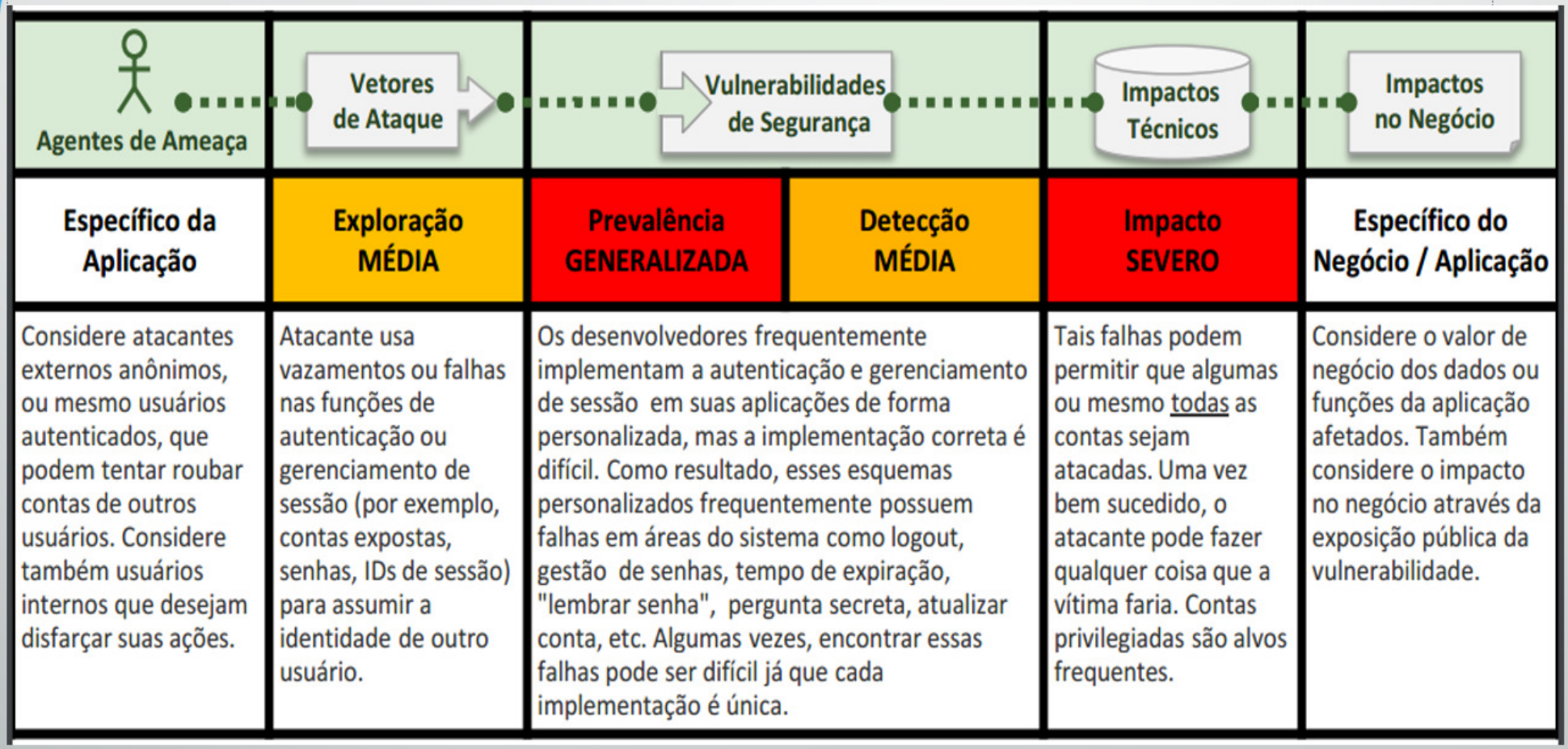
Fatea 03/09/2016

Segurança.

O que é?

- Segundo a OWASP (Open Web Application Security Project / Projeto de Segurança de Aplicações Open Web) .

- Funções da aplicação relacionadas com autenticação e gerenciamento de sessão geralmente são implementadas de forma incorreta, permitindo que os atacantes comprometam senhas, chaves e tokens de sessão ou, ainda, explorem outra falha da implementação para assumir a identidade de outros usuários.

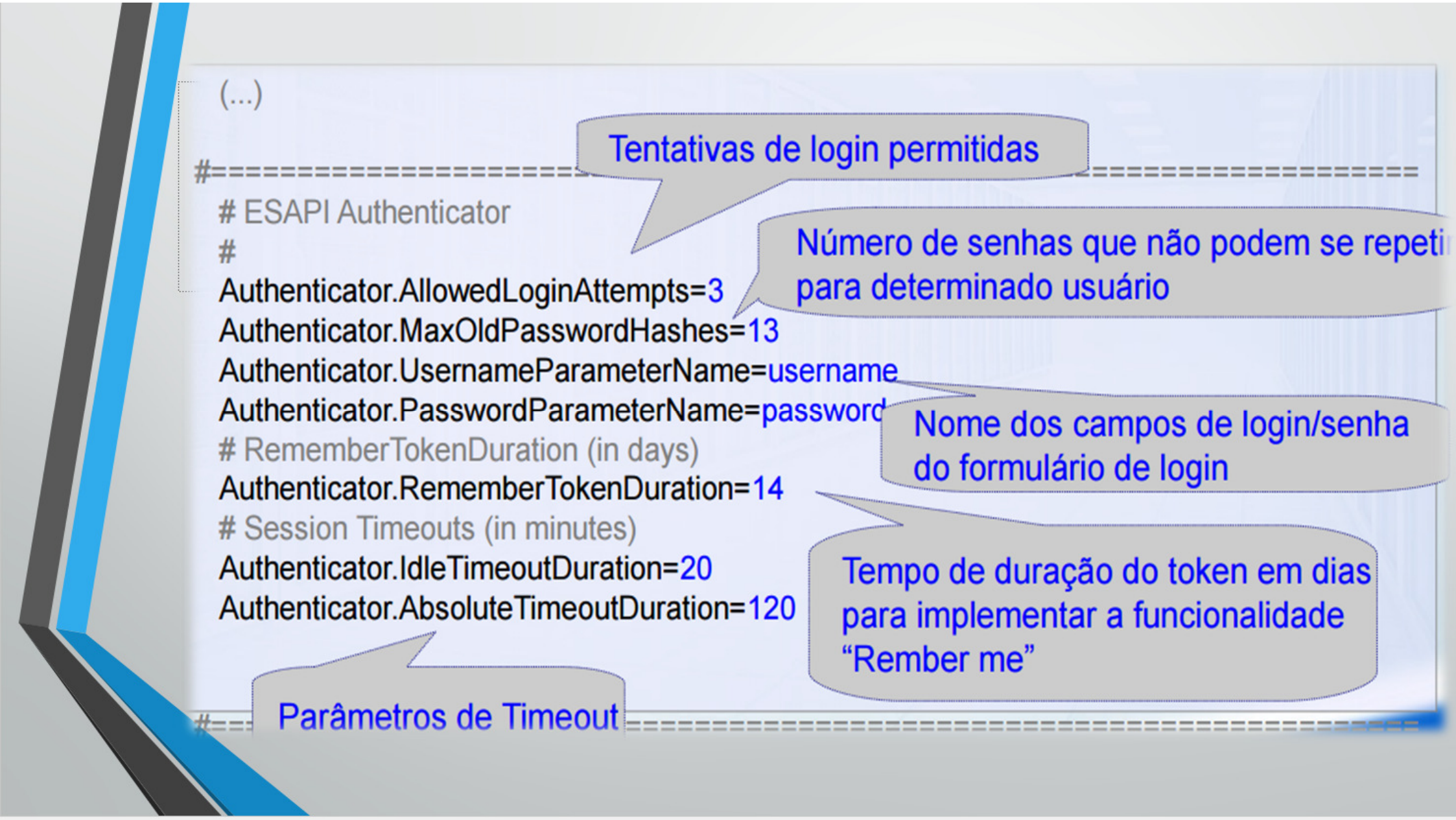

Agentes de Ameaça
Vetores de Ataque
Vulnerabilidades de Segurança
Impactos Técnicos
Impactos no Negócio
Específico da Aplicação
Exploração MÉDIA
Prevalência GENERALIZADA
Detecção MÉDIA
Impacto SEVERO
Específico do Negócio / Aplicação
Considere atacantes externos anônimos, ou mesmo usuários autenticados, que podem tentar roubar contas de outros usuários. Considere também usuários internos que desejam disfarçar suas ações.
Atacante usa vazamentos ou falhas nas funções de autenticação ou gerenciamento de sessão (por exemplo, contas expostas, senhas, IDs de sessão) para assumir a identidade de outro usuário.
Os desenvolvedores frequentemente implementam a autenticação e gerenciamento de sessão em suas aplicações de forma personalizada, mas a implementação correta é difícil. Como resultado, esses esquemas personalizados frequentemente possuem falhas em áreas do sistema como logout, gestão de senhas, tempo de expiração, "lembrar senha", pergunta secreta, atualizar conta, etc. Algumas vezes, encontrar essas falhas pode ser difícil já que cada implementação é única.
Tais falhas podem permitir que algumas ou mesmo todas as contas sejam atacadas. Uma vez bem sucedido, o atacante pode fazer qualquer coisa que a vítima faria. Contas privilegiadas são alvos frequentes.
Considere o valor de negócio dos dados ou funções da aplicação afetados. Também considere o impacto no negócio através da exposição pública da vulnerabilidade.

Estou Vulnerável ?

- Os ativos de gerenciamento de sessão, como credenciais do usuário e IDs de sessão, são protegidos adequadamente? Você pode estar vulnerável se:
- 1. As credenciais de autenticação de usuário não estão protegidas utilizando hash ou criptografia, quando armazenadas.
- 2. As credenciais podem ser descobertas através de fracas funções de gerenciamento de contas (por exemplo, criação de conta, alteração de senha, recuperação de senha, IDs de sessão fracos).
- 3. IDs de sessão são expostos na URL (por exemplo, reescrita de URL).
- 4. IDs de sessão são vulneráveis a ataques de fixação de sessão.
- 5. IDs de sessão não expiram, ou sessões de usuário ou tokens de autenticação, particularmente aqueles baseados em single sign-on (SSO), e não são devidamente invalidados durante o logout.
- 6. IDs de sessão não são rotacionados após o login bem-sucedido.
- 7. Senhas, IDs de sessão, e outras credenciais são enviadas através de conexões não criptografadas

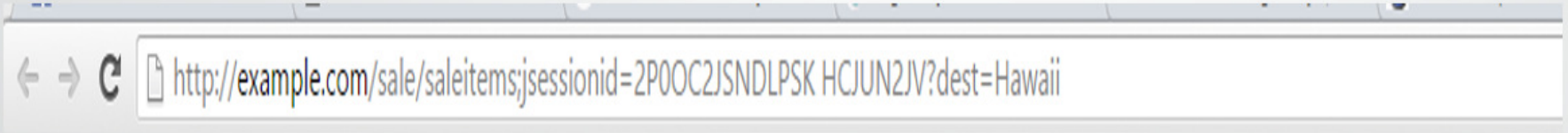
Como evito ?

- A recomendação principal para uma organização é disponibilizar aos desenvolvedores:
- 1. Um conjunto único de controles fortes de autenticação e gerenciamento de sessão. Tais controles devem procurar:
- a) Cumprir todos os requisitos de autenticação e gerenciamento de sessão definidos no Padrão de Verificação de Segurança da Aplicação do OWASP (ASVS). (Visto nesse link https://www.owasp.org/images/b/b3/OWASP_SCP_v1.3_pt-BR.pdf).
- b) ter uma interface simples para os desenvolvedores. Considere a biblioteca do ESAPI Authenticator e User APIs como base.


(...)
#==========
ESAPI Authenticator
#
Authenticator.AllowedLoginAttempts=3
Authenticator.MaxOldPasswordHashes=13
Authenticator.UsernameParameterName=username
Authenticator.PasswordParameterName=password
# RememberTokenDuration (in days)
Authenticator.RememberTokenDuration=14
# Session Timeouts (in minutes)
Authenticator.IdleTimeoutDuration=20
Authenticator.AbsoluteTimeoutDuration=120
#==========
Tentativas de login permitidas
Número de senhas que não podem se repeti
para determinado usuário
Nome dos campos de login/senha
do formulário de login
Tempo de duração do token em dias
para implementar a funcionalidade
“Rember me”
Parâmetros de Timeout

Exemplos de ataque.

- 1: Uma aplicação de reservas de passagens aéreas suporta reescrita de URL, colocando IDs de sessão na URL:



- Um usuário autenticado do site quer deixar seus amigos saberem sobre a venda. Ele envia um e-mail do link acima sem saber que com isso também está enviando a sua ID da sessão. Quando seus amigos utilizarem o link, irão usar sua sessão e cartão de crédito. Cenário #
- 2: O tempo de expiração da aplicação não está definido corretamente. O usuário utiliza um computador público para acessar o site. Em vez de selecionar "logout" o usuário simplesmente fecha a aba do navegador e vai embora. O atacante usa o mesmo navegador uma hora mais tarde, e esse navegador ainda está autenticado. Cenário #
- 3: Atacante interno ou externo ganha acesso ao banco de dados de senhas do sistema. Senhas de usuários não estão utilizando hash, expondo assim todas as senhas dos usuários ao atacante.

Fontes

- http://virtx.com.br/seguranca-na-internet-os-10-riscos/
- http://softwarelivre.gov.br/palestras-tecnicas-cisl/palestra2_spb
- https://www.owasp.org/images/9/9c/OWASP_Top_10_2013_PT-BR.pdf
- http://softwarelivre.gov.br/palestras-tecnicas-cisl/material_complementar_spb